

Claim Amendment Summary

Claims pending

- At time of the Action: Claims 1-48.
- After this Response: Claims 1, 4-26, 28-30, 32-42, and 44-48.

Canceled or Withdrawn claims: 2, 3, 27, 31, and 43.

Amended claims: 1, 4, 11, 19, 25, 30, 36, 41, 42, 47, and 48.

New claims: none

Please amend claims 1, 11, 19, 25, 30, 36, 41, 42, 47, and 48 as follows:

1. **(CURRENTLY AMENDED)** In a paging operating system having physical memory for holding information and secondary storage comprising a page file for receiving information that is paged out from the physical memory, a computer-implemented method of protecting information comprising:

creating a key and page locking the key in the physical memory, wherein creating the key comprises creating the key during system boot up, wherein different keys can be created during different system boot ups;

encrypting information using the ~~[[a]]~~ key ~~that is page locked in the physical memory;~~ and

paging out, to the page file, the encrypted information.

2-3. **(CANCELED)**

4. **(CURRENTLY AMENDED)** The computer-implemented method of claim 1 [[2]], wherein said creating the key further comprises generating a random key with a random key generator.

1 5. **(ORIGINAL)** The computer-implemented method of claim 4,
2 wherein said generating comprises using RSA RC4 as an encryption algorithm to
3 generate the key.

4
5 6. **(ORIGINAL)** The computer-implemented method of claim 1,
6 wherein said encrypting comprises:

7 calling an operating system kernel;

8 the kernel using the page-locked key to encrypt the information.

9
10 7. **(ORIGINAL)** The computer-implemented method of claim 6,
11 wherein said calling is performed by an application.

12
13 8. **(ORIGINAL)** The computer-implemented method of claim 6,
14 wherein said calling is performed by an operating system memory manager.

15
16 9. **(ORIGINAL)** One or more computer-readable media having
17 computer-readable instructions thereon which, when executed by a computer,
18 perform the computer-implemented method of claim 1.

19
20 10. **(ORIGINAL)** An operating system programmed with instructions
21 which, when implemented by the operating system, implement the method of
22 claim 1.

23
24 11. **(CURRENTLY AMENDED)** In a paging operating system having
25 main memory for holding information and secondary storage comprising a page

1 file for receiving information that is paged out from the main memory, a
2 computer-implemented method of protecting information comprising:

3 creating a key during system boot up, wherein different keys can be created
4 during different system boot ups;

5 page-locking the [[a]] key in main memory;
6 restricting access to the page-locked key to only the operating system
7 kernel;
8 calling the operating system kernel to encrypt information;
9 accessing the page-locked key with the operating system kernel; and
10 using the operating system kernel to encrypt the information with the page-
11 locked key.

12
13 12. (ORIGINAL) The computer-implemented method of claim 11,
14 wherein said calling is performed by an operating system memory manager.

15
16 13. (ORIGINAL) The computer-implemented method of claim 11,
17 wherein said calling is performed by an application.

18
19 14. (ORIGINAL) The computer-implemented method of claim 11
20 further comprising prior to said calling:

21 designating at least one page in the main memory with a designation;
22 recognizing the designation and, responsive thereto, calling the operating
23 system kernel to encrypt the information.

24
25 15. (ORIGINAL) The computer-implemented method of claim 14,
wherein said recognizing is performed by the memory manager.

1
2 16. **(ORIGINAL)** The computer-implemented method of claim 11,
3 wherein said calling comprises specifying a memory location and a memory size
4 associated with the information to be encrypted.

5
6 17. **(ORIGINAL)** One or more computer-readable media having
7 computer-readable instructions thereon which, when executed by a computer,
8 perform the computer-implemented method of claim 11.

9
10 18. **(ORIGINAL)** An operating system programmed with instructions
11 which, when implemented by the operating system, implement the method of
12 claim 11.

13
14 19. **(CURRENTLY AMENDED)** In a paging operating system having
15 main memory for holding information and secondary storage comprising a page
16 file for receiving information that is paged out from the main memory, a
17 computer-implemented method of handling encrypted information comprising:

18 accessing encrypted information in the page file; and

19 decrypting the encrypted information with a key created during system boot
20 up, wherein different keys can be created during different system boot ups and
21 wherein the key that is page-locked in the main memory.

22
23 20. **(ORIGINAL)** The computer-implemented method of claim 19
24 further comprising placing the decrypted information in a page of main memory.
25

1 21. **(ORIGINAL)** The computer-implemented method of claim 19
2 further comprising placing the decrypted information in a page-locked page of
3 main memory.

4
5 22. **(ORIGINAL)** The computer-implemented method of claim 19,
6 wherein the page-locked key is accessible only to the operating system kernel.

7
8 23. **(ORIGINAL)** One or more computer-readable media having
9 computer-readable instructions thereon which, when executed by a computer,
10 perform the computer-implemented method of claim 19.

11
12 24. **(ORIGINAL)** An operating system programmed with instructions
13 which, when implemented by the operating system, implement the method of
14 claim 19.

15
16 25. **(CURRENTLY AMENDED)** In a paging operating system having
17 main memory for holding information and secondary storage comprising a page
18 file for receiving information that is paged out from the main memory, a
19 computer-implemented method of protecting information comprising:

20 allocating a non-pageable page of main memory during system boot;

21 generating a random key, wherein different keys can be generated during
22 different system boots; and

23 storing the random key in the non-pageable page of main memory, the
24 random key being configured for use by the operating system to encrypt
25 information that might be paged out to the page file.

1 26. **(ORIGINAL)** The computer-implemented method of claim 25,
2 wherein said generating comprises using an RSA RC4 encryption algorithm.

3
4 27. **(CANCELED)**

5
6 28. **(ORIGINAL)** One or more computer-readable media having
7 computer-readable instructions thereon which, when executed by a computer,
8 perform the computer-implemented method of claim 25.

9
10 29. **(ORIGINAL)** An operating system programmed with instructions
11 which, when implemented by the operating system, implement the method of
12 claim 25.

13
14 30. **(CURRENTLY AMENDED)** In an operating system having main
15 memory for holding information and secondary storage for receiving information
16 that is transferred out of main memory, a computer-implemented method of
17 protecting information comprising:

18 generating at least one non-pageable random key by using a random key
19 generation process during system boot up, wherein different keys can be generated
20 during different system boot ups;

21 encrypting at least one selected block of information in the main memory
22 with a software component that uses the at least one random key for encryption;

23 transferring the one encrypted block of information to the secondary
24 storage;

25 decrypting the one encrypted block of information with the software
component that uses the at least one random key for decryption; and

1 placing the decrypted block of information in the main memory.

2
3 31. **(CANCELED)**

4
5 32. **(ORIGINAL)** The computer-implemented method of claim 30
6 further comprising restricting access to the at least one random key to only the
7 software component.

8
9 33. **(ORIGINAL)** The computer-implemented method of claim 30,
10 wherein the software component comprises the operating system's kernel.

11
12 34. **(ORIGINAL)** The computer-implemented method of claim 30
13 further comprising:

14 storing the at least one random key in the main memory; and

15 locking the at least one random key in the main memory so that it does not
16 get transferred to the second storage.

17
18 35. **(ORIGINAL)** An operating system programmed with instructions
19 which, when implemented by the operating system, implement the method of
20 claim 30.

21
22 36. **(CURRENTLY AMENDED)** A system for use in protecting
23 pageable information comprising:

24 a memory having pageable and non-pageable pages; and
25

1 at least one key created during system boot and stored in the memory in a
2 non-pageable page, the key being configured for use in encrypting pageable
3 information, wherein different keys can be created during different system boots.
4

5 37. (ORIGINAL) The system of claim 36 further comprising a software
6 component that is configured to access and use said one key to encrypt pageable
7 information.
8

9 38. (ORIGINAL) The system of claim 37, wherein the one key is
10 accessible only to the software component.
11

12 39. (ORIGINAL) The system of claim 37 further comprising at least
13 one application configured to call the software component to encrypt the pageable
14 information.
15

16 40. (ORIGINAL) The system of claim 37 further comprising a memory
17 manager configured to call the software component to encrypt the pageable
18 information.
19

20 41. (CURRENTLY AMENDED) A computer program embodied on
21 one or more computer-readable media, the program comprising:

22 creating a key and page locking the key in main memory of a computer,
23 wherein creating the key comprises creating the key during system boot up,
24 wherein different keys can be created during different system boot ups;
25

1 encrypting information with the ~~[[a]]~~ key ~~that is page-locked in main~~
2 ~~memory of a computer;~~
3 paging out, to secondary storage, the encrypted information;
4 accessing the encrypted information in the secondary storage; and
5 decrypting the encrypted information with the key that is page-locked in the
6 main memory.

7
8 42. (CURRENTLY AMENDED) A programmable computer
9 comprising:

10 a processor;
11 main memory for holding information;
12 secondary storage for receiving information that is temporarily transferred
13 out of the main memory;

14 the computer being programmed with computer-readable instructions
15 which, when executed by the processor, cause the computer to:

16 generate a key during system boot up, wherein different keys can be
17 generated during different system boot ups;

18 page lock the key in the main memory;

19 encrypt information that is to be transferred to the secondary storage with
20 the ~~[[a]]~~ key ~~that is locked in the main memory;~~

21 transfer the encrypted information to the secondary storage; and

22 decrypt the encrypted information with a key that is locked in the main
23 memory.

24
25 43. (CANCELED)

1 44. **(ORIGINAL)** The programmable computer of claim 42, wherein
2 the key that is used to encrypt the information is the same key that is used to
3 decrypt the information.

4
5 45. **(ORIGINAL)** The programmable computer of claim 42, further
6 comprising a software component that is programmed to encrypt and decrypt the
7 information.

8
9 46. **(ORIGINAL)** The programmable computer of claim 45, wherein
10 the software component comprises the operating system's kernel.

11
12 47. **(CURRENTLY AMENDED)** One or more application
13 programming interfaces embodied on one or more computer-readable media for
14 execution on a computer in conjunction with a paging operating system having
15 main memory for holding information and a page file for receiving information
16 that is paged out from the main memory, comprising:

17 an interface method for generating a key during system boot up, wherein
18 different keys can be generated during different system boot ups;

19 an interface method for page locking the key in the main memory,

20 an interface method for encrypting pageable information with the [[a]] key
21 ~~that is page-locked in the main memory;~~ and

22 an interface method for decrypting encrypted information that is contained
23 in the page file.

24
25 48. **(CURRENTLY AMENDED)** An application programming
interface embodied on a computer-readable medium for execution on a computer

1 in conjunction with a paging operating system having main memory for holding
2 information and secondary storage comprising a page file for receiving
3 information that is paged out from the main memory, comprising a method for
4 setting an attribute on a page of main memory, the attribute designating that the
5 page must be encrypted with a key created during system boot up and that is page-
6 locked in the main memory prior to the page being paged out to the page file,
7 wherein different keys can be created during different system boot ups.
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25